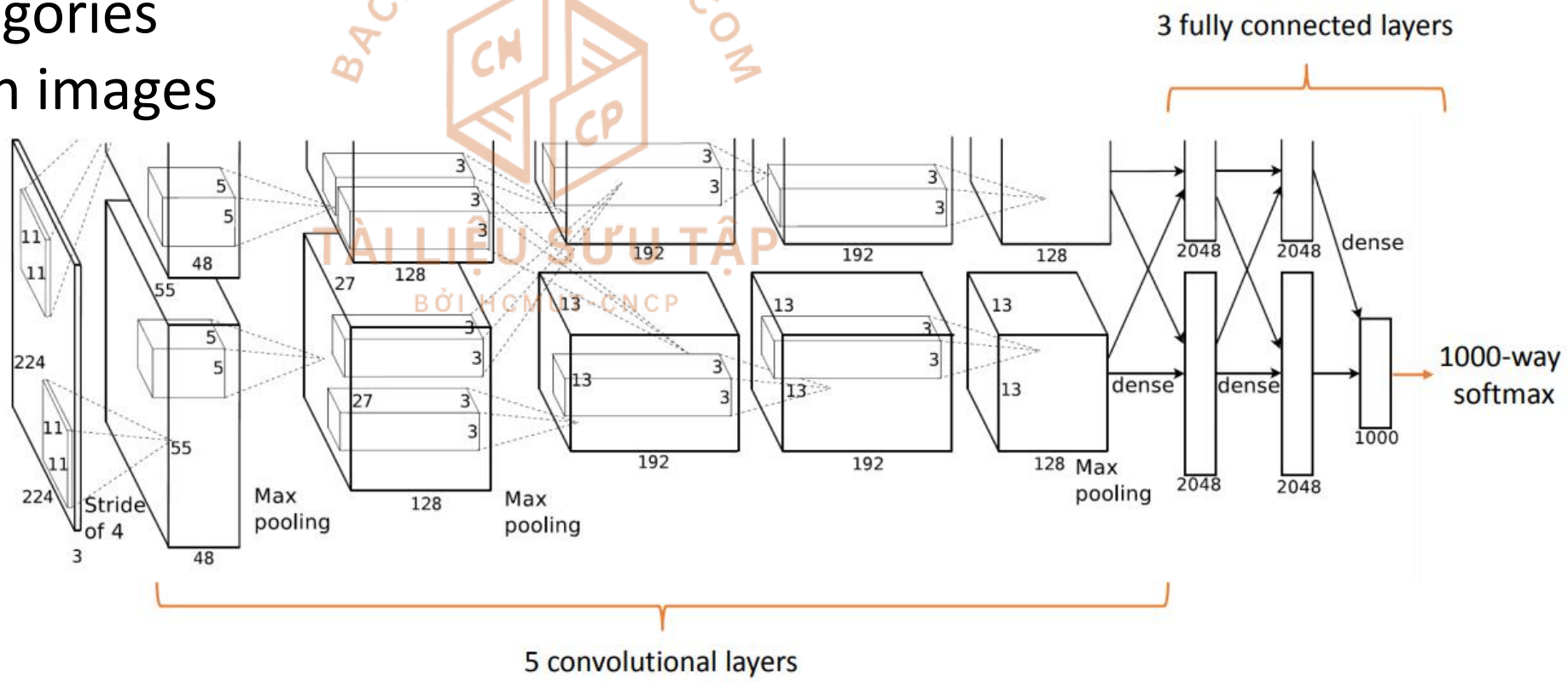
Artificial Intelligence

Softmax function

Pham Viet Cuong
Dept. Control Engineering & Automation, FEEE
Ho Chi Minh City University of Technology

Softmax function

- ✓ Usual output layer for classification tasks
- ✓ Example: Image classification problem ILSVRC2012 (Large Scale Visual Recognition Challenge 2012)
  - ❖ 1000 categories
  - ❖ 1.2 million images



Softmax function

- Ordinary classifier:
  - Some rule, or function, that assigns to a sample x a class label $\hat{y}$

$$\hat{y} = f(x)$$

- Probabilistic classifier:
  - Conditional distributions $P(\mathbf{Y} \mid \mathbf{X})$: for a given $x \in \mathbf{X}$, assign probabilities to all $y \in \mathbf{Y}$ (these probabilities sum to one).

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0.20 \\ 0.35 \\ 0.05 \\ 0.40 \end{bmatrix}$$

Softmax function

- Example: Image classification problem
  - 4 categories: dog, cat, kite, car
  - One-hot encoding

$$\begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \quad \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \quad \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

Input → Black box → Output

$$\mathbf{z} = \begin{bmatrix}z_1\\z_2\\z_3\\z_4\end{bmatrix} = \begin{bmatrix}0.8\\1.4\\-0.8\\0.2\end{bmatrix} \xrightarrow{\text{Softmax}} \mathbf{y} = \begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix} = \begin{bmatrix}0.28\\0.51\\0.06\\0.15\end{bmatrix}$$

Softmax function

- Softmax function:
  - Takes as input a vector of *k* real numbers
  - Normalizes it into a probability distribution consisting of *k* probabilities proportional to the exponentials of the input numbers

$$\begin{bmatrix} 0.8 \\ 1.4 \\ -0.8 \\ 0.2 \end{bmatrix}$$

- Input:
  - Some vector components could be negative, or greater than one
  - Might not sum to 1
- Output:
  - Each component in the interval (0,1)
  - The components add up to 1
  - Larger input components correspond to larger probabilities
  - Can be interpreted as probabilities

$$\begin{bmatrix} 0.28 \\ 0.51 \\ 0.06 \\ 0.15 \end{bmatrix}$$

Softmax function

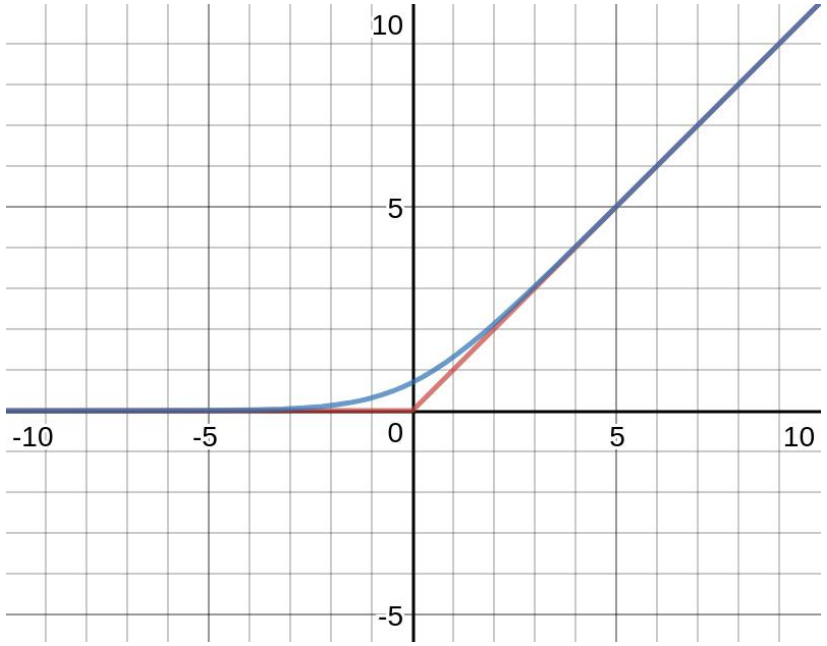
- Softmax function:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \forall i = 1, 2, .., C$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1.4 \\ -0.8 \\ 0.2 \end{bmatrix} \Rightarrow \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0.28 \\ 0.51 \\ 0.06 \\ 0.15 \end{bmatrix}$$

Softmax function

- Where comes the name “softmax”?
  - Smooth approximation to the maximum function?
  - Smooth approximation to the arg max function --> **softargmax**



$$f(x) = \max(0, x)$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1.4 \\ -0.8 \\ 0.2 \end{bmatrix} \quad \arg\max(\mathbf{z}) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0.28 \\ 0.51 \\ 0.06 \\ 0.15 \end{bmatrix}$$

Softmax function

- References
  - https://cs231n.github.io/linear-classify/#softmax
  - http://neuralnetworksanddeeplearning.com/chap3.html
  - https://en.wikipedia.org/wiki/Softmax_function
  - https://en.wikipedia.org/wiki/Probabilistic_classification
  - A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, 2012
  - https://www.thehappycatsite.com/funny-cat-quotes/